

# 高效的 H. 264 并行编码算法

孙书为, 陈书明

(国防科技大学计算机学院, 湖南长沙 410073)

**摘 要:** CABAC 是 H. 264/AVC 视频压缩标准主要档次中采用的熵编码机制, 结合 RDO 模式选择技术, 可以降低 20% 的编码码率, 但是编码器计算复杂度却同时大大增加. 对算法进行并行化是有效加快编码速度的方法, 但是, 由于 CABAC 具有自适应编码的特点和 RDO 模式选择对熵编码的使用, 使得顺序编码的宏块之间存在着严格的数据相关性, 限制了并行编码算法的开发. 本文结合基于宏块区域划分的数据级并行编码机制 MBRP 和码率估计技术, 为采用 CABAC 熵编码机制的 H. 264 编码算法提供了一种高效的并行编码方案: 将 H. 264 编码算法划分为模式选择和码流生成两个部分, 使之构成典型的生产者-消费者关系; 将 RDO 模式选择中的 CABAC 替换为码率估计, 去除模式选择过程中因 CABAC 导致的严格数据相关性; 对模式选择部分采用 MBRP 并行机制; 码流生成部分由单独的处理完成, 并和模式选择部分实现流水线并行处理. 通过 4 处理器系统模拟器进行实验, 发现在保持视频压缩性能几乎不变的情况下, 该并行算法的加速比可以达到 4.7.

**关键词:** H. 264; CABAC; 码率估计; RDO 模式选择; 多处理器; 并行算法; 宏块区域划分

**中图分类号:** TN919. 8 **文献标识码:** A **文章编号:** 0372-2112 (2009) 02-0357-05

## An Efficient Parallel Algorithm for H. 264/AVC Encoder

SUN Shu-wei, CHEN Shu-ming

(Computer School, National University of Defense Technology, Changsha, Hunan 410073, China)

**Abstract:** CABAC is the entropy coding method adopted in H. 264/AVC main profile, and the combination of RDO and CABAC can represent an improvement in bit rate of around 20%. However, the computational complexity increases drastically as well. Parallel algorithms are efficient ways to speed up the encoder. However, because of the adaptive characteristic of CABAC, there are strict data-dependencies between macro-blocks encoding, which makes it hard to exploit the parallelism of encoder. In this paper, an efficient parallel algorithm for H. 264/AVC encoder with CABAC entropy coding is proposed based on the Macro-Block Region Partition (MBRP) parallel method and CABAC bit-rate estimation technique: the H. 264 encoder is partitioned into the mode decision part and the bit-stream producing part with relationship of producer and consumer; CABAC in the RDO mode decision is replaced with bit-rate estimation, which removes the data-dependencies brought by CABAC; in the mode decision part, the MBRP parallel algorithm is adopted; the bit-stream producing part is performed in a single processor, and forms task pipelining with mode decision part. Simulation results on 4-processor simulator show that the proposed parallel algorithm can achieve speedups above 4.7 while maintaining the similar R-D performance.

**Key words:** H. 264; CABAC; bit-rate estimation; RDO (rate-distortion optimization) mode decision; multi-processor; parallel algorithm; MBRP (macro-block region partition)

### 1 引言

H. 264/AVC 是 ITU-T 和 MPEG 联合制定的一个国际性视频编解码标准, 它以传统的混合编码框架为基础, 提供了更多可供选择的编码模式和更加高效的熵编码机制 - CABAC (上下文自适应二进制算术编码 Context-Based Adaptive Binary Arithmetic Coding)<sup>[1]</sup>, 从而获得了更高的视频压缩率.

为了在多种编码模式中选择出最好的一种, 人们提出了 RDO (率失真优化) 模式选择方法<sup>[2]</sup>, 以穷举方式计算编码块在每一种模式下的率失真性能, 选择具有最小 Lagrange 耗费的编码模式作为最佳模式, 如式 (1) 所示. 此处, 为 Lagrange 算子, 和量化参数  $QP$  相关;  $D$  是重构失真;  $R$  是熵编码后的码流位数.

$$J = D + \lambda R \quad (1)$$

文献[3]研究了 RDO 和 CABAC 的性能, 发现 RDO

和 CABAC 的结合可以降低 20 % 的码率, 但是其代价是计算复杂度的明显增加, 算术编码和 RDO 模式选择的计算时间大约占总编码时间的 50 %.

为了减少视频编码算法的计算复杂度, 人们在快速模式选择算法方面做了大量的研究工作<sup>[4~8]</sup>. 但是, 对于 H. 264 编码算法的实时应用而言仍然是远远不够的. 为了获得更高的编码性能, 一些硬件编码器设计完成<sup>[9~12]</sup>, 但是由于硬件编码器不能适应算法的改进, 因此其应用受到很大的限制.

随着 VLSI 和 SoC 技术的不断进步和日益成熟, 多处理器系统日益成为微处理器设计领域的一个重要方向. 而将高计算复杂度的 H. 264 编码算法进行并行化以加快编码速度成为一种合理的选择. 人们基于不同的应用平台, 提出了多种方式(比如任务流水, 线程级并行和不同粒度的数据级并行等)的并行算法<sup>[13~16]</sup>. 然而, 现有的并行算法都是针对 H. 264 基本档次开发的, 并不能直接应用在主要档次中. 由于主要档次中采用的 CABAC 熵编码机制具有自适应编码特性, 导致在熵编码过程中存在严格的数据相关性, 这对并行算法的开发而言几乎是不可克服的障碍.

在先前的研究中, 我们提出了一种码率估计技术可以对 CABAC 熵编码位数进行准确的估计<sup>[17]</sup>. 本文结合 MBRP 数据级并行机制和码率估计技术, 为采用 CABAC 熵编码机制的 H. 264 编码算法提供了一种高效的并行编码方案. 通过 4 处理器系统模拟器进行实验, 发现在对视频压缩性能影响不大的基础上, 该并行方案可以显著提高 H. 264 视频编码的速度.

## 2 MBRP 并行编码机制

在 H. 264 编码器中, 视频图像首先划分为  $16 \times 16$  的宏块, 数据以宏块为单位按光栅扫描顺序依次进行处理. 通过对 H. 264 视频编码算法进行分析, 可以发现在 H. 264 编码算法中存在 3 种数据相关性: (1) 帧间数据相关性, 宏块只有在其需要的参考重构数据都已经可见时才可以开始处理; (2) 帧内行间数据相关性, 宏块只有在其上方 3 个相邻宏块已经编码并重构后才可以开始处理; (3) 帧内行内数据相关性, 宏块只有在其左侧相邻宏块已经编码并重构后才可以开始处理<sup>[16]</sup>.

根据上述 3 种数据相关性, 我们提出了 MBRP 并行编码算法<sup>[16]</sup>:

(1) 根据 H. 264 编码算法的编码参数和实际多处理器系统的处理器数目确定视频图像的宏块区域划分, 使得每个宏块区域尽量包含相同数目的宏块列.

(2) 将不同的宏块区域分配给不同的处理器进行编码, 在启动多处理器系统执行算法时, 采用 wave-front 技术, 依次延迟一定的时间, 回避帧内行内数据相关

性, 如图 1 所示(单斜线区域为已编码宏块, 双斜线区域为当前编码宏块).

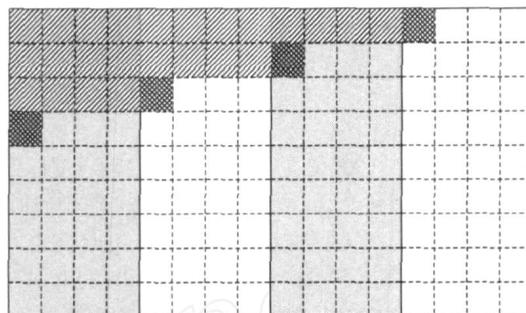


图1 宏块区域划分和MBRP并行算法

(3) 处理器在编码过程中, 以宏块为单位按光栅扫描顺序处理宏块区域; 处理器编码完边界宏块(宏块区域的第一列宏块和最后一列宏块)后, 将相关的重构数据, 相关宏块的运动向量和非 0 变换量化系数的数目值传送给相邻的处理器.

(4) 宏块区域完成去块效应滤波后将滤波后的重构数据传送给所有的处理器.

(5) 在不存在数据相关的情况下, 所有的处理器并行工作, 处理视频图像中不同宏块区域的数据.

然而, 上述 MBRP 并行编码算法只适用于 H. 264 基本档次. 由于在 H. 264 主要档次采用的熵编码机制 CABAC 中, 每编码一个二进制符号后都会对相应语法单元的概率统计进行更新, 因此顺序编码的二进制符号(相同语法单元)在熵编码过程中存在严格的数据相关性. 此时上述‘帧内行间数据相关性’变为宏块只有在上方所有宏块已经编码后才可以开始处理, 这就使得宏块必须以光栅扫描顺序依次进行处理.

在 H. 264 编码算法中, CABAC 主要有 2 个应用: 在 RDO 模式选择中获得编码码率; 在最终熵编码过程中生成码流. 其中 RDO 模式选择在整个视频编码算法中所占的比重非常大, 超过 90 % (包括运动估计); 而熵编码比重则较小, 不到 1 %<sup>[3]</sup>. 由于在 RDO 模式选择过程中真正需要的是编码位数而不是码流本身, 因此, 如果我们可以通过估计的方法获得码率就可以避免在 RDO 模式选择中采用 CABAC, 也就避免了编码过程中的严格数据相关性, 从而为开发模式选择部分的数据级并行性扫清障碍.

## 3 CABAC 码率估计技术

CABAC 是一种简化的算术编码机制, 和 RDO 结合可以有效地提高视频压缩的效率<sup>[18]</sup>. 我们在先前的研究中讨论了 RDO 模式选择过程中 CABAC 熵编码机制下的码率估计技术<sup>[17]</sup>, 以此为基础, 我们给出 MBRP 并行机制下的码率估计方法.

假设二进制符号“0”的概率为  $P_{ZERO}$ , 那么熵编码 1

位“0”和1位“1”分别可以生成  $B_{ZERO}$  和  $B_{ONE}$  位码流:

$$B_{ZERO} = \ln P_{ZERO} / \ln 0.5 \quad (2)$$

$$B_{ONE} = \ln (1 - P_{ZERO}) / \ln 0.5 \quad (3)$$

在计算机上实现对数函数是非常复杂的工作,因此我们采用简化的方法获得  $B_{ZERO}$  和  $B_{ONE}$  的值。“0”的概率  $P_{ZERO} \in [0, 1]$ , 我们将该区间划分为若干个子区间,对每个子区间取其中间值作为该子区间的典型概率,并以此典型概率获得  $B_{ZERO}$  和  $B_{ONE}$  的值应用于整个子区间,表 1 给出了一种子区间划分的例子。

表 1 概率区间的划分及其  $B_{LBS}$  和  $B_{MBS}$  值

$P_{ZERO}$	$B_{ZERO}$	$B_{ONE}$	$P_{ZERO}$	$B_{ZERO}$	$B_{ONE}$
0.05	4.322	0.074	0.55	0.862	1.152
0.10	3.322	0.152	0.60	0.737	1.322
0.15	2.737	0.234	0.65	0.621	1.515
0.20	2.322	0.322	0.70	0.515	1.737
0.25	2.000	0.415	0.75	0.415	2.000
0.30	1.737	0.515	0.80	0.322	2.322
0.35	1.515	0.621	0.85	0.234	2.737
0.40	1.322	0.737	0.90	0.152	3.322
0.45	1.152	0.862	0.95	0.074	4.322
0.50	1.000	1.000	/	/	/

假设编码某一语法单元时,需要编码“0”的数目为  $N_{ZERO}$ ,需要编码“1”的数目为  $N_{ONE}$ ,那么编码该语法单元需要的码流位数可以估计为

$$B_{ESTI} = B_{ZERO} \times N_{ZERO} + B_{ONE} \times N_{ONE} \quad (4)$$

在第  $th$  种模式下编码块的总位数  $B_T(QP, )$  包括三部分:量化变换系数的编码位数  $B_C(QP, )$ ,运动向量的编码位数  $B_M(QP, )$  和头信息(包括编码模式和编码块类型(CBP)等信息)的编码位数  $B_H(QP, )$ :

$$B_T(QP, ) = B_C(QP, ) + B_M(QP, ) + B_H(QP, ) \quad (5)$$

根据对 CABAC 熵编码的描述<sup>[18]</sup>,  $B_C(QP, )$  部分包括 6 种不同的编码符号,分别为:

$$\begin{aligned} & coded\_block\_flag, \quad significant\_coeff\_flag, \\ & last\_significant\_coeff\_flag, \quad coeff\_greater\_one\_flag, \\ & coeff\_abs\_level\_minus2, \quad coeff\_sign\_flag. \end{aligned}$$

本文以这 6 种符号为单位估计  $B_C(QP, )$  部分的码率。通过对量化变换系数块进行扫描,可以获得如下信息:  $Num_{nonzero\_coeff}$ , 系数块中非 0 系数的数目;  $Num_{one\_coeff}$ , 系数块中绝对值等于 1 的系数数目;  $Pos_{last\_coeff}$ , 最后一个非 0 系数的位置;  $Sum_{absminus2\_coeff}$ , 所有非 0 且绝对值大于 1 的系数绝对值减 2 的和。通过这些信息,可以得到上述 6 种编码的  $N_{ZERO}$  和  $N_{ONE}$ , 比如

$significant\_coeff\_flag$ , 每个编码块中包含  $Num_{nonzero\_coeff}$  个“1”和  $Pos_{last\_coeff} - Num_{nonzero\_coeff}$  个“0”。

$B_M(QP, )$  部分包括 3 种不同的编码符号,分别为:  $significant\_mvd\_flag$ ,  $mvd\_abs\_level\_minus1$  和  $mvd\_sign\_flag$ 。本文以这 3 种符号为单位估计  $B_M(QP, )$  部分的码率。通过对当前编码模式的每一对运动向量值进行扫描,可以得到如下信息:  $Num_{zero\_mvd}$ , 等于 0 的运动向量分量数目;  $Num_{nonzero\_mvd}$ , 不等于 0 的运动向量分量数目;  $Sum_{absminus1\_mvd}$ , 所有非 0 运动向量分量的绝对值减 1 的和。通过这些信息,可以得到上述 3 种编码符号的  $N_{ZERO}$  和  $N_{ONE}$ , 比如  $mvd\_abs\_level\_minus1$ , 每个帧间预测编码块会出现  $Num_{nonzero\_mvd}$  个“0”和  $Sum_{absminus1\_mvd}$  个“1”。

累计当前帧已编码块中每种编码符号的  $N_{ZERO}$  和  $N_{ONE}$  可以得到该编码符号已经编码的“0”和“1”总数目,进而获得“0”的概率  $P_{ZERO}$ , 通过表 1 得到  $B_{ZERO}$  和  $B_{ONE}$  值。此时,编码块中  $B_C(QP, )$  和  $B_M(QP, )$  部分每种编码符号的  $B_{ZERO}$ ,  $B_{ONE}$  和  $N_{ZERO}$ ,  $N_{ONE}$  都已知,可以通过式(4)获得其码率估计值。值得注意的是,在顺序编码的算法中,当前帧已编码块为按照光栅扫描顺序当前块以前的所有块,而在 MBRP 并行算法中,已编码块为如图 1 中单斜线区域的宏块,因此上述两种算法进行估计码率时用到的  $B_{ZERO}$  和  $B_{ONE}$  并不一致。

对  $B_H(QP, )$  部分,由于其在整个码流中所占的比重较小,我们采用较为简单的方法进行码率估计,比如对编码模式,我们直接采用编码模式的索引值作为其码率估计值,通过实验发现编码效果是可以接受的。

#### 4 基于码率估计技术的快速模式选择算法

结合上面两节的技术,我们给出 CABAC 熵编码器制下的 H.264 并行编码方案 MBRP+ 如下:

(1) 将 H.264 编码算法划分为模式选择和码流生成两个部分,其中后者在最优模式下进行熵编码从而生成码流,而前者包括除此之外所有的编码功能。

(2) 将模式选择部分的 CABAC 替换为码率估计,采用估计的编码位数计算候选模式的 Lagrange 耗费,并以此为依据进行模式选择。

(3) 采用 MBRP 机制对模式选择部分进行并行化处理,具体算法如下:

(a) 确定视频图像的宏块区域划分,使得每个宏块区域尽量包含相同数目的宏块列;

(b) 将不同宏块区域的模式选择分配给不同的处理器处理,在启动多处理器系统时,依次延迟一定的时间,回避帧内行内数据相关性;

(c) 每个处理器在编码一个宏块区域的过程中,以

宏块为单位顺序进行,每个处理器完成边界宏块的模式选择后,将相关的重构数据和运动向量传送给相邻的处理器;

(d) 处理器完成一个宏块的模式选择后,将当前宏块各种编码符号的“0”和“1”数目传输给其他所有的处理器以更新其  $P_{ZERO}$ ;

(e) 处理器完成其宏块区域的去块效应滤波后,将滤波后的重构数据传送给所有的处理器;

(f) 在不存在数据相关的情况下,所有的处理器并行工作,完成不同宏块区域的模式选择。

(4) 码流生成部分由单独的处理器完成以维护其严格的数据相关性。

(5) 对模式选择部分和码流生成部分进行流水化处理,并行执行以进一步加快算法的执行速度。

## 5 实验结果与分析

为验证本文提出的 H.264 并行编码算法 MBRP+ 的有效性,我们基于 JMI0.2<sup>[19]</sup> 开发了 H.264 编码器并行算法的软件模拟器:模拟器设置为 4 处理器系统;将视频图像划分为 4 个近似相等的宏块区域,每个处理器处理一个宏块区域的模式选择部分;由于码流生成部分在整个编码器中的比重较小,将其安排在第一个处理器上执行,即第一个处理器在执行完其负责的宏块区域所有宏块的模式选择后,执行整帧图像的熵编码过程,生成最终的码流;处理器间采用消息传递机制传输相关数据。

为此,我们修改了 JMI0.2 中的宏块编码函数  $encode\_one\_macroblock()$ ,为其增加结构化变量参数  $enc\_picture$ 。该变量用于存放不同宏块区域的原始数据、重构数据和其他编码过程中生成的中间数据。视频帧编码过程依然按照光栅扫描顺序进行,但对位于不同宏块区域的宏块,调用  $encode\_one\_macroblock()$  函数时采用不同的  $enc\_picture$  参数(模拟不同处理器处理不同宏块区域的过程)。由于不同宏块区域的原始数据和重构数据位于不同的  $enc\_picture$  变量中,视频帧编码过程中的数据相关性决定了必须在不同的  $enc\_picture$  变量之间进行数据的复制(模拟相关数据交换的过程)。

编码过程中统计每一个宏块的编码时间,同时根据该宏块的编码启动时间计算其编码结束时间。对宏块区域中非边界宏块,其编码启动时间一般为其左侧宏块的编码结束时间。对边界宏块,其编码启动时间是所有相关宏块编码结束时间加上相关数据传输时间(不同  $enc\_picture$  变量间数据的复制)中的最大值。这样可以获得每一个宏块区域的编码结束时间(宏块区域最后一个宏块的编码结束时间)。通过统计宏块熵编码函数  $write\_one\_macroblock()$ ,可以获得视频帧的熵

编码总时间。将第一个宏块区域的编码结束时间加上熵编码总时间就得到了并行编码时间。

为了便于比较,本文共模拟了 3 种不同的算法: MBRP+ 并行编码算法, JMI0.2 串行编码算法和基于码率估计的快速模式选择算法(FMDA, 仅以 CABAC 码率估计替代 RDO 模式选择中的熵编码)。实验中编码器参数设置为:采用全搜索运动估计算法,单参考帧,帧间预测搜索范围为 16;采用 Hadamard 变换;采用 RDO 模式选择;采用 CABAC 熵编码模式;编码序列为 IPPPP;量化参数为 28,32,36 和 40。

实验结果以性能加速比  $R$ 、码率差异  $Bit-rate$  和峰值信噪比差异  $PSNR$  的形式在表 2 中给出,所有的差异值均是相对于 JMI0.2 的结果获得的。

表 2 CIF 格式视频编码实验结果

视频序列	算法	$R_{Ave}$	$PSNR$	$Bit-rate$
News 300 帧	MBRP+	4.509	-0.073 dB	+1.281 %
	FMDA	1.429	-0.052 dB	+1.065 %
Foreman 300 帧	MBRP+	4.478	-0.062 dB	+1.554 %
	FMDA	1.436	-0.059 dB	+1.460 %
Coast Guard 300 帧	MBRP+	4.663	-0.074 dB	+1.829 %
	FMDA	1.484	-0.066 dB	+1.733 %
Mobile 300 帧	MBRP+	5.064	-0.047 dB	+0.603 %
	FMDA	1.619	-0.039 dB	+0.597 %
Tempete 260 帧	MBRP+	4.798	-0.063 dB	+1.086 %
	FMDA	1.522	-0.051 dB	+0.996 %
平均	MBRP+	4.702	-0.064 dB	+1.271 %
	FMDA	1.498	-0.053 dB	+1.170 %

实验中统计的  $PSNR$  值是亮度和色度的平均  $PSNR$  值,计算过程参见文献[4]。 $Bit-rate$  表示相对于 JMI0.2 的码率改变量的百分比,  $PSNR$  表示平均  $PSNR$  值的改变量,计算过程参见文献[20],此处不再赘述。性能加速比  $R$  由式(6)定义:

$$R = \frac{T_{originalRDO}}{T_{proposed}} \quad (6)$$

其中,  $T_{originalRDO}$  是原始 H.264 编码器(本文中采用 JMI0.2)在进行 RDO 模式选择情况下的编码时间,  $T_{proposed}$  是进行 RDO 模式选择并采用码率估计方法和并行算法的编码时间。 $R_{Ave}$  取值为 4 种量化参数下加速比  $R$  的平均值。

表 2 中的数据说明,本文提出的 H.264 并行编码方案 MBRP+ 具有良好的性能,和 JMI0.2 相比,在对率失真性能影响不大的情况下,获得了较为理想的性能加速比(其中部分加速效果是采用码率估计技术获得的)。图 2 给出了 Foreman(CIF)序列通过 JMI0.2 和本文

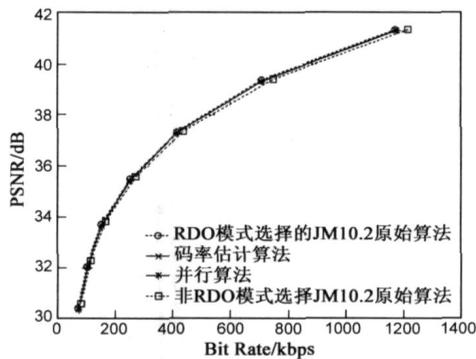


图2 Foreman(CIF)序列的率失真曲线

提出的算法进行编码后获得的率失真曲线,该图说明 MBRP+ 和 JM10.2 具有近似的率失真性能。

## 6 结论

CABAC 和 RDO 的结合进一步改进了视频编码算法的压缩性能,但是也使得算法复杂度大大增加。为了加快视频编码的速度,本文结合 MBRP 并行编码机制和 CABAC 码率估计技术,给出了一种 CABAC 熵编码机制下的 H.264 并行编码方案 MBRP+。通过 4 处理器系统模拟器进行实验发现,MBRP+ 并行编码方案在对率失真性能影响不大的情况下,获得了较为理想的性能加速比。

### 参考文献:

- [1] ISO/IEC FDIS 14 496-10, Information Technology - Coding of Audio-Visual Objects - Part 10: Advanced Video Coding, Final Draft International Standard[S]. 2003.
- [2] G Sullivan, T Wiegand, K-P Lim. Joint model reference encoding methods and decoding concealment methods [A]. 9th JVT Meeting (JVT-I049d0) [C]. San Diego, 2003.
- [3] J-L Nunez- Yanez, V-A Chouliaras. Hardware assisted rate distortion optimization with embedded CABAC accelerator for the H.264 advanced video codec [J]. IEEE Transactions on Consumer Electronics, 2006, 52(2): 590 - 597.
- [4] F Pan, X Lin. Fast mode decision algorithm for intra-prediction in H.264 video coding [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2005, 15(7): 813 - 822.
- [5] 李世平, 蒋刚毅, 郁梅. 快速帧内预测模式选择新方法 [J]. 电子学报, 2006, 34(1): 141 - 146.  
Li Shiping, Jiang Gangyi, Yu Mei. New fast mode selection for intra prediction [J]. Acta Electronica Sinica, 2006, 34(1): 141 - 146. (in Chinese)
- [6] 贾克斌, 谢晶, 方湘. 一种基于自相关法的 H.264 高效帧内预测算法 [J]. 电子学报, 2006, 34(1): 152 - 154.  
Jia Kebin, Xie Jing, Fang Xiang. An efficient intra frame pre-

diction algorithm based on autocorrelation for H.264/AVC [J]. Acta Electronica Sinica, 2006, 34(1): 152 - 154. (in Chinese)

- [7] 宋彬, 常义林, 周宁兆. 基于 H.264 帧间预测的快速算法 [J]. 电子学报, 2006, 34(1): 31 - 34.  
Song Bin, Chang Yilin, Zhou Ningzhao. Fast inter prediction method based H.264 [J]. Acta Electronica Sinica, 2006, 34(1): 31 - 34. (in Chinese)
- [8] Z-P Lin, H-T Yu, F Pan. A scalable fast mode decision algorithm for H.264 [A]. Proceedings of the 2006 International Symposium on Circuits and Systems [C]. Island of Kos: IEEE Circuits and Systems Society, 2006. 3494 - 3497.
- [9] Y-W Huang, B-Y Hsieh, T-C Chen, L-G Chen. Analysis, fast algorithm and VLSI architecture design for H.264/AVC intra frame coder [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2005, 15(3): 378 - 401.
- [10] T-C Chen, S-Y Chien, Y-W Huang, C-H Tsai, C-Y Chen, T-W Chen, L-G Chen. Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2006, 16(6): 673 - 688.
- [11] M Sayed, I Amer, W Badawy. Towards an H.264/AVC full encoder on chip: an efficient real-time VBSME ASIC chip [A]. Proceedings of the 2006 International Symposium on Circuits and Systems [C]. Island of Kos: IEEE Circuits and Systems Society, 2006. 2613 - 2616.
- [12] C-C Cheng, C-W Ku, T-S Chang. A 1280x720 pixels 30 frames/s H.264/MPEG-4 AVC intra encoder [A]. Proceedings of the 2006 International Symposium on Circuits and Systems [C]. Island of Kos: IEEE Circuits and Systems Society, 2006. 5335 - 5338.
- [13] A Gulati, G Campbell. Efficient mapping of the H.264 encoding algorithm onto multi-processor DSPs [A]. Proceedings of SPIE-IS&T Electronic Imaging [C]. San Jose, California: SPIE, 2005. 94 - 103.

### 作者简介:



孙书为 男, 1979 年生于山东临清, 2003 年毕业于国防科技大学计算机学院, 获工学硕士学位, 现为国防科技大学计算机学院博士生, 主要研究方向为视频编解码算法和高性能微处理器设计。  
E-mail: sws97@163.com

陈书明 男, 1961 年生于安徽六安, 1993 年毕业于国防科技大学计算机系, 获工学博士学位, 现为国防科技大学计算机学院教授, 博士生导师, 主要研究方向为高性能微处理器设计和超深亚微米 VLSI 设计理论与技术。E-mail: smchen@nudt.edu.cn (下转第 392 页)

Packaging Technologies, 2008, 31(4): 890 - 903.

- [12] Yang Chun-xin, Zhou Jian-hui. Multi-disciplinary design optimization of the plate fin heat sink[A]. Thermal Challenges in

Next Generation Electronic Systems [C]. Millpress. Rotterdam, 2007. 343 - 350.

#### 作者简介:



周建辉 男, 1976 年生于江西. 北京航空航天大学人机与环境工程专业博士. 主要研究方向为计算传热学和流体力学等.

Email: yzhou130@126.com

杨春信 男, 1965 生于湖南. 北京航空航天大学人机与环境工程专业博士生导师. 主要研究方向为多相流传热、飞机环境控制与安全救生、人机与环境工程等.

(上接第 361 页)

- [14] Y-K Chen, X Tian, S Ge, M Girkar. Towards efficient multi-level threading of h. 264 encoder on intel hyper-threading architectures[A]. Proceedings of 18th International Parallel and Distributed Processing Symposium[C]. Santa Fe, New Mexico: IEEE Computer Society, 2004.
- [15] Z Zhao, P Liang. Data partition for wave-front parallelization of H. 264 video encoder[A]. Proceedings of the 2006 International Symposium on Circuits and Systems[C]. Island of Kos: IEEE Circuits and Systems Society, 2006. 2669 - 2672.
- [16] S-W Sun, D Wang, S-M Chen. A highly efficient parallel algorithm for H. 264 encoder based on macro-block region partition[A]. Proceedings of the 2007 International Conference on High Performance Computing and Communications[C]. Houston: Springer, 2007. 577 - 585.
- [17] 孙书为, 陈书明, 刘祥远. CABAC 熵编码机制下有效的

变换域码率估计技术[J]. 电子学报, 2008, 36(8): 1512 - 1518.

Sun Shuwei, Chen Shuming, Liu Xiangyuan. Efficient Transform domain bit-rate estimation technique for CABAC[J]. Acta Electronica Sinica, 2008, 36(8): 1512 - 1518. (in Chinese)

- [18] D Marpe, H Schwarz, T Wiegand. Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard[J]. IEEE Transaction on Circuits and Systems for Video Technology, 2003, 13(7): 620 - 636.
- [19] JM10.2. Reference Software of JVT[DB/OL]. <http://bs.hhi.de/suehring/tml/download/jm10.2.zip>, 2006 - 01 - 06
- [20] G Bjontegaard. Calculation of average PSNR differences between RD-curves (VCEG-M33) [A]. 13th VCEG Meeting [C]. Austin, 2001.